1. Content-Addressable Memory (CAM) Fundamentals

CAM is a specialized type of memory used in high-speed searching applications. While standard SRAM reads or writes data based on a given address, CAM performs the inverse operation.

- **Operation:** The system provides "data" (a search key) to the CAM, and the CAM searches its contents to see if that data exists.
- **Parallel Search:** The matching operation happens simultaneously across the memory array. If the input key exists in the CAM, a "matchline" output is asserted for the specific word containing that key.
- **Result:** If a match is found, the active matchline can be used to access a corresponding RAM location to retrieve associated data. If no match is found, a "miss" signal is generated.

2. Ternary CAM (TCAM) for Networking

While standard CAMs are useful for exact matches, networking hardware often requires more flexibility.

- **Binary Limitation:** Standard CAMs can only store and match logic **0** and logic **1**.
- **The TCAM Solution:** A Ternary CAM (TCAM) can store three values: **0**, **1**, and **X** ("Don't Care").
- **Network Application:** This "Don't Care" state is essential for network routers.
    - Routers often route data based on a subnet address (e.g., the first 24 bits of an IP) while ignoring the specific host address bits.
    - TCAMs enable the router to match the prefix while treating the remaining bits as "don't care," allowing for efficient packet forwarding.

3. TCAM Storage Logic

To store three distinct states (0, 1, X), a TCAM cell requires more complex hardware than a standard binary memory cell.

- **Two-Bit Encoding:** A TCAM cell uses two bits of state (labeled A and B) to represent the three values.

- **State Definitions:**
  - **Match 0:** When A=1 and B=0, the cell matches an input of **0**.
  - **Match 1:** When A=0 and B=1, the cell matches an input of **1**.
  - **Don't Care (X):** When A=0 and B=0, the cell matches **both** 0 and 1.

4. Hardware Characteristics and Constraints

For Information Security and Networking professionals, understanding the physical constraints of CAM/TCAM explains certain limitations in hardware performance and cost.

- **Physical Size:** CAM cells are significantly larger than standard memory cells, generally occupying about twice the area of an SRAM cell.
- **Power Consumption:** CAMs consume relatively large amounts of power.
  - This is because the matchlines are heavily loaded and have a high activity factor (close to 1) during search operations.
  - To reduce power, some designs separate the search lines from the bitlines to lower capacitance.
- **Signal Timing:** To ensure reliability, CAM arrays often use strobe signals to drive wordlines only after matchlines have settled, thereby preventing errors during the high-speed search process.

12.6 Content-Addressable Memory

Figure 12.67 shows the symbol for a content-addressable memory (CAM). The CAM acts as an ordinary SRAM that can be read or written given *adr* and *data*, but also performs *matching* operations. Matching asserts a *matchline* output for each word of the CAM that contains a specified *key*.

A common application of CAMs is translation lookaside buffers (TLBs) in microprocessors supporting virtual memory. The virtual address is given as the key to the TLB CAM. If this address is in the CAM, the corresponding matchline is asserted. This matchline can serve as the wordline to access a RAM containing the associated physical address, as shown in Figure 12.68. A NOR gate processing all of the matchlines generates a *miss* signal for the CAM. Note that the *read*, *write*, and *adr* lines for updating the TLB entries are not drawn.

Figure 12.69 shows a 10T CAM cell consisting of a normal SRAM cell with additional transistors to perform the match. Multiple CAM cells in the same word are tied to the same matchline. The matchline is either precharged or pulled high as a distributed pseudo-nMOS gate. The key is placed on the bitlines. If the key and the value stored in the cell differ, the matchline will be pulled down. Only if all of the key bits match all of the bits stored in the word of memory will the matchline for that word remain high. The key can contain a "don't care" by setting both *bit* and *bit_b* low. The inside front cover shows a layout of this cell in a $56 \times 43\ \lambda$ area; CAMs generally have about twice the area of SRAM

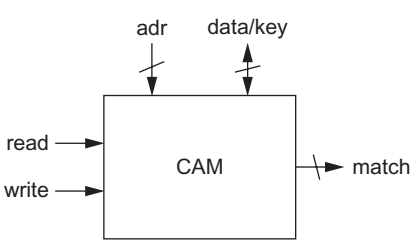


FIGURE 12.67 Content-addressable memory

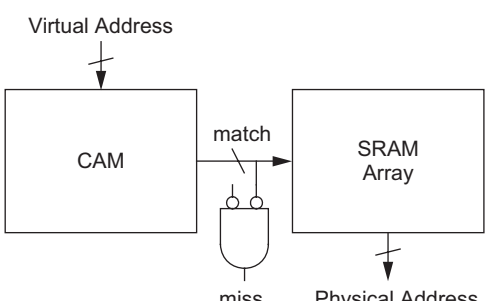


FIGURE 12.68 Translation Lookaside Buffer (TLB) using CAM

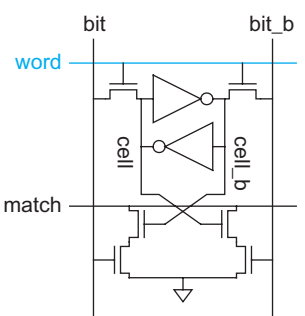


FIGURE 12.69 CAM cell implementation

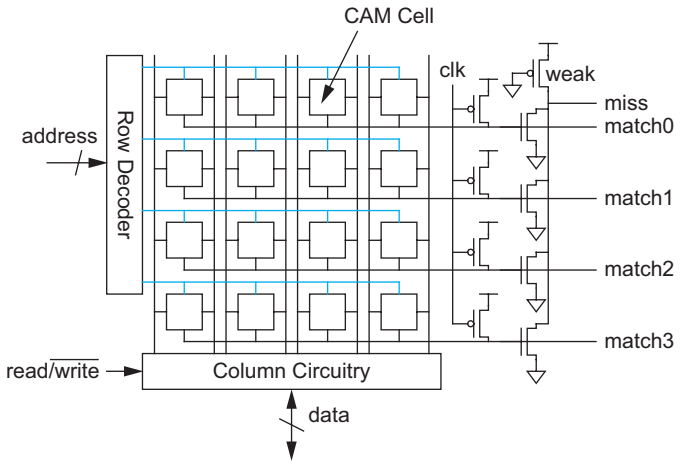


FIGURE 12.70 4 × 4 CAM array

cells. Sometimes the key is provided on separate *searchlines* rather than on the bitlines to reduce the capacitance and power consumption of a search.

Figure 12.70 shows a complete 4 × 4 CAM array. Like an SRAM, it consists of an array of cells, a decoder, and column circuitry. However, each row also produces a dynamic matchline. The matchlines are precharged with the clocked pMOS transistors. The *miss* signal is produced with a distributed pseudo-nMOS NOR.

When the matchlines are used to access a RAM, the monotonicity problem must be considered. Initially, all the matchlines are high. During CAM operation, the lines pull down, leaving at most one line asserted to indicate which row contains the key. However, the RAM requires a monotonically rising wordline. Figure 12.71 refines Figure 12.68 with strobed AND gates driving the wordlines as early as possible after the matchlines have settled. The strobe can be timed with an inverter chain or replica delay line in much the same way that the sense amplifier clock for an SRAM was generated in Section 12.2.3.3. As usual, self-timing margin must be provided so the circuit operates correctly across all design corners. The strobe must be deasserted before the match lines precharge.

In some applications, a CAM doesn't care about the value of certain bits. For example, a CAM used in a network router may not care about the subnet address when it is seeking to route data to the correct continent. A *ternary CAM* (TCAM) can store *X* (don't care) values as well as 0 and 1 bits. Figure 12.72 shows a TCAM cell using two bits of state to store the three values. This cell also illustrates separating the search lines from the bitlines. When $A = 1$ and $B = 0$, the cell matches a 0. When $A = 0$ and $B = 1$, the cell matches a 1. When $A = 0$ and $B = 0$, the cell matches both 0 and 1.

Large CAMs can use many of the same techniques as large RAMs, including sense amplifiers and multiple subarrays. They tend to consume relatively large amounts of power because the matchlines are heavily loaded and have an activity factor close to 1. [Pagiamtzis06] surveys many alternative CAM structures such as NAND architectures. [Agrawal08] offers a power and delay model.

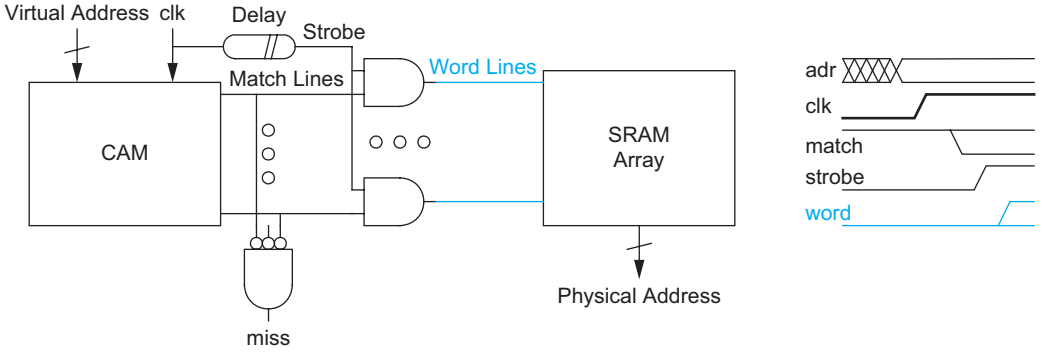


FIGURE 12.71 Refined TLB path with monotonic wordlines

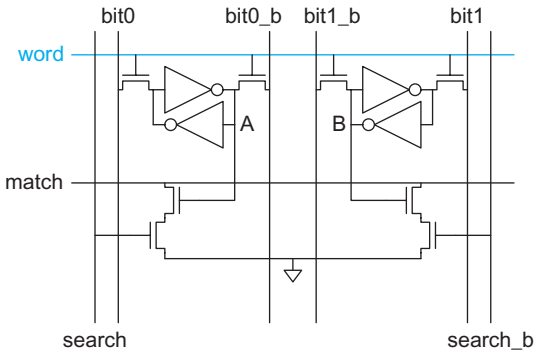


FIGURE 12.72 TCAM